ESCOLA SUPERIOR DE TECNOLOGIA DE TOMAR

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

INTRODUÇÃO À PROGRAMAÇÃO

(ENG. INFORMÁTICA)

COMPUTADORES E PROGRAMAÇÃO I

(ENG. ELECTROTÉCNICA)

2002/2003

TRABALHO PRÁTICO Nº 5

Cotação: **0.25 Valores**

Objectivos :

- Modularização.
- Criação de bibliotecas de funções.

Exercícios:

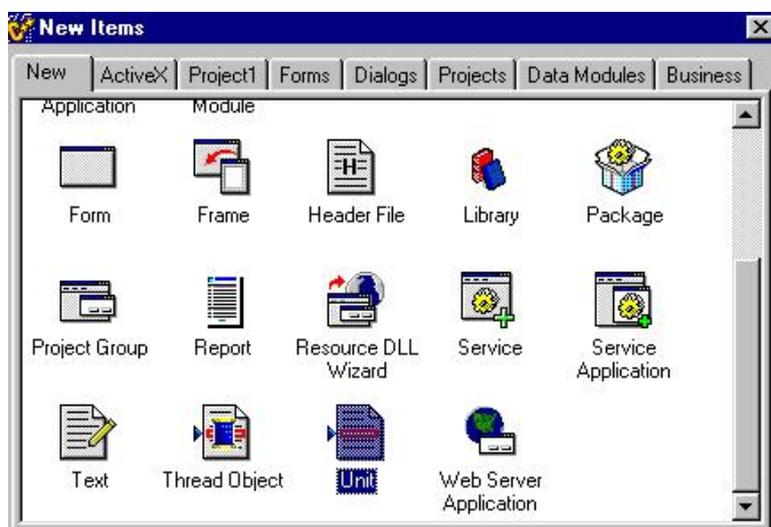
No projecto de software é usual estarem envolvidas equipas constituídas por diversas pessoas. Ocupando-se cada uma delas por desenvolverem pequenas partes do projecto total. Ou seja a modularização do problema é fundamental.

Nesta aula prática vamos aprender a criar bibliotecas de funções que poderão vir a ser usadas mais tarde na resolução de outros problemas.

Comece por executar o C++ Builder, criando um projecto do tipo console wizard.

Guarde o projecto com o nome exercicio1.
Este projecto, para além dos ficheiros habituais inclui mais dois ficheiros, o matematica.c e o matematica.h. Estes ficheiros são usados para guardar um conjunto de funções ao qual se dá o nome de biblioteca de funções. Neste caso vamos dar o nome matemática à biblioteca que vamos criar.

Criação da biblioteca matemática

- File -> New -> Unit

- File - > Save **Matematica.c**

Guarde a nova Unit com o nome **matematica.c.** As bibliotecas de funções são compostas por dois ficheiros, um com a extensão **.c** e outro com a extenção **.h**. No ficheiro que tem a extensão **.c** guardamos as funções com o código que fonte que as concretiza. No ficheiro **.h** guardamos os protótipos das funções.

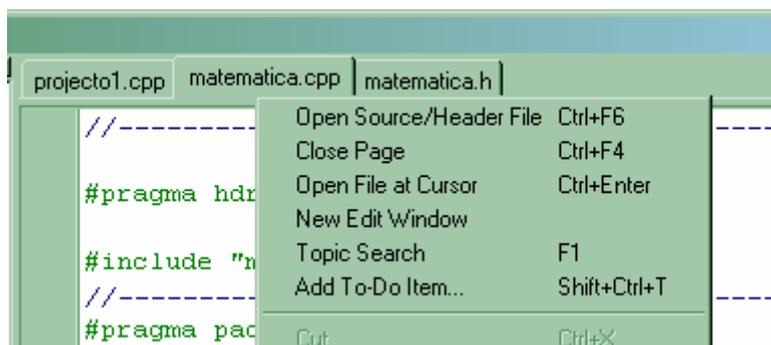
Por exemplo:

Matemática.c	Matemática.h

int soma(int A, int B)
{
int resultado;
resultado=A+B;
return resultado;
//código da função soma
}
	int soma(int A, int B);
	//protótipo da função soma

No exemplo anterior mostramos como incluir na biblioteca matemática a função soma. Repare, que os programadores que vão usar as funções da biblioteca matemática (neste caso apenas está definida a função soma) apenas necessitam de conhecer o conteúdo do ficheiro .h. Senão vejamos, no ficheiro .h está definido o nome da função (soma), os parâmetros de entrada e o seu tipo (int A e int B) e o tipo de dados de saída (int). O ficheiro .h funciona como um manual de instruções para a biblioteca de funções matemática. Podemos, por exemplo vender a biblioteca matemática com o conteúdo do ficheiro .c compilado sem dar a conhecer os algoritmos usados na concepção das funções da biblioteca.

Neste momento apenas está visível o ficheiro .c, precisamos de abrir o **matematica.h** onde se vai especificar o conjunto de protótipos das funções da biblioteca.

- Carregar no botão direito em cima do ficheiro **matematica.c** escolher **Open Source/header file** e guardar o ficheiro.

Agora já temos os três ficheiros necessários, só precisamos de escrever o código em cada um deles e executar o programa

1. Construa um programa que peça um valor real ao utilizador e imprima o valor da sua exponencial (e^x) calculado a partir da fórmula de Taylor.

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \ldots$$

   Adicione á biblioteca **matematica** as funções:

   - Factorial -> int factorial(int n);
   - Exponencial -> float exponencial(float x);

   e utilize-a para realizar o cálculo.

2. Construa um programa que:

   Calcule o valor do seno pela fórmula $senx = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots$. Para tal construa uma função chamada seno, pertencente à sua biblioteca matemática.

   Imprima a diferença entre a função seno escrita na alínea anterior e a função sin pertencente à biblioteca math. Protótipo da função seno -> float seno(float x);

3. Acrescente à biblioteca matemática uma função que determina se um número é primo. Teste a função. Protótipo da função -> bool n_perfeito(int n);

4. Acrescente à biblioteca matemática uma função que determine se um número inteiro é perfeito. Um número é perfeito se a soma dos seus divisores for igual a ele próprio. Teste a função. Protótipo da função primo bool primo(int n);

5. Pretende-se construir uma aplicação que apresente um menu através do qual vão ser seleccionados várias operações e uma opção para sair.

   - Construa uma biblioteca chamada **Menu** e implemente as seguintes funções:
     - **void linha(int largura, char limite);**
       - Desenha uma linha com o carácter “limite” de tamanho “largura”.
     - **void interior(int largura, char limite, char fundo);**
       - Desenha uma linha tem “largura” caracteres, em que o primeiro e o último são “limite” e os restantes “fundo”
     - **void caixa(int largura, int altura, char limite, char fundo);**
       - Utiliza as funções anteriores para desenhar uma caixa com dimensão “largura” x “altura” com o caracter “limite” na fronteira e o carácter “fundo” no espaço interior

- o **void desenhaMenu();**
  - ▪ Desenha uma caixa e imprime o texto com as opções disponíveis.
- o **char escolheMenu();**
  - ▪ Desenha o menu e solicita ao utilizador que escolha uma opção válida.

- Construa o programa principal que solicita a escolha de uma opção do menu. O programa termina quando se escolher a opção “sair” e imprime a mensagem “modulo não disponível” para todas as outras.